

REMARKS

This Amendment is submitted in response to the Examiner's Action mailed September 28, 2004, with a shortened statutory period of three months set to expire December 28, 2004. Claims 1-45 are currently pending. With this amendment, claims 1, 3-6, 8, 12, 14-16, 18-21, 23, 27, 29-31, 33-36, 38, 42, and 44-45 have been amended, and claims 2, 13, 17, 28, 32, and 43 have been canceled.

The Examiner stated that should claims 2 and 17 be found to be allowable, claims 13 and 28 would be objected to under 37 C.F.R. 1.75 as being a substantial duplicate thereof. Applicants have canceled claims 2, 17, and 32. Therefore, no future objection under 37 C.F.R. 1.75 is necessitated.

Applicants have amended the claims to describe the logically partitioned data processing system including a plurality of partitions. Each partition executes its own separate operating system. Errors from an operating system being executed by one partition do not affect the operation of the operating systems being executed by the other partitions.

An unrecoverable error is encountered in one of the processors that is assigned to one of the partitions. The error results in a crash of the partition that includes the processor that had the error. Remaining partitions continue to function after the crash. Each processor is assigned to a different one of the partitions.

Trace data is stored in the same trace buffer for all of the partitions. Data that is already stored in the trace buffer is continually being overwritten with newly received trace data as the newly received trace data is received within the trace buffer. Error data associated with the error is stored in the trace buffer. The current contents of the trace buffer are saved prior the error data being overwritten.

The Examiner rejected claims 1-11, 13-14, 16-26, 28-29, 31-41, 43, and 44 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent 6,493,837 issued to *Pang* in view of U.S. Patent 6,543,010 issued to *Gaudet*. This rejection, as it might be applied to the claims as amended, is respectfully traversed.

The Examiner states that *Pang* does not teach the events and the logs being preserved in the event of a partition crash. The Examiner uscs *Gaudet* to supply this missing feature.

The Examiner states that *Pang* teaches a logically partitioned data processing system by teaching an association of buffers with processors. Applicants have amended the claims to define a logically partitioned data processing system as being a system where each partition executes its own separate operating system, and errors from an operating system being executed by one partition do not affect the operation of the operating systems being executed in the other partitions. *Pang* does not teach such a logically partitioned data processing system. The association of buffers with processors is not a system that includes partitions where each partition executes its own separate operating system, and errors from an operating system being executed by one partition do not affect the operation of the operating systems being executed in the other partitions.

*Gaudet* also does not teach a logically partitioned data processing system. *Gaudet* does not teach a system where each partition executes its own separate operating system, and errors from an operating system being executed by one partition do not affect the operation of the operating systems being excuted in the other partitions.

Applicants claim the trace data from all partitions being stored in the same trace buffer. *Pang* does not teach a trace buffer where data from all partitions are stored in the same trace buffer. *Pang* teaches multiple log buffers where one or more log buffers are associated with just one processor. See Column 5, lines 7-10. Thus, *Pang* does not teach multiple processors being associated with one log buffer. In *Pang*, the data associated with one processor is stored in one or more log buffers that are associated with that one processor. The data associated with multiple processors are not stored together in one log buffer. Therefore, *Pang* does not teach a trace buffer where data from all partitions are stored in the same trace buffer.

Applicants claim continually overwriting the data in the trace buffer with newly received data as the newly received data is received within the buffer. *Pang* does not teach the data that is already stored in a log buffer being continually overwritten with newly received data as the newly received data is received within the buffer. In *Pang*, when a log buffer becomes full, its association with its processor is removed and its

contents are stored. Thus, the data in a buffer in *Pang* cannot be overwritten because it is removed to permanent storage after the log buffer becomes full and before there is a chance the data might be overwritten. Therefore, *Pang* does not teach continually overwriting the data in the trace buffer with newly received data as the newly received data is received within the buffer.

Applicants claim saving the current contents of the trace buffer prior to the error data being overwritten. The data in a buffer in *Pang* cannot be overwritten because it is removed to permanent storage after the log buffer becomes full or when a maintenance thread executes. The Examiner states that *Pang* teaches, in column 5, lines 11-25, storing contents of a trace buffer prior to the data being overwritten. This section of *Pang* teaches the data being copied from a log buffer to permanent storage after the log buffer becomes full. The data may also be copied from a log buffer before the buffer becomes full as part of the execution of a maintenance thread. However, nothing in *Pang* describes the possibility that data may be overwritten. Thus, *Pang* does not teach saving the current contents of the trace buffer prior to the error data being overwritten.

*Pang* does not teach a logically partition data processing system, storing trace data from all partitions in the same trace buffer, continually overwriting data that is already stored in the trace buffer with newly received data, or saving the current contents of the trace buffer prior to error data in the buffer being overwritten.

The Examiner states that *Pang* does not teach the events being errors and the logs being preserved in the event of a partition crash. The Examiner relies on *Gaudet* to supply the features missing from *Pang*. As discussed above, *Pang* does not teach a logically partition data processing system, storing trace data from all partitions in the same trace buffer, continually overwriting data that is already stored in the trace buffer with newly received data, or saving the current contents of the trace buffer prior to error data in the buffer being overwritten.

*Gaudet* teaches detecting an error, and in response to the error, read at least a portion of the data stored in system memory, compressing the data, and storing the compressed data on a storage device. *Gaudet* does not teach a logically partition data processing system, storing trace data from all partitions in the same trace buffer, continually overwriting data that is already stored in the trace buffer with newly received

data, storing error data associated with the error in the trace buffer, or saving the current contents of the trace buffer prior to error data in the buffer being overwritten.

The combination of *Pang* and *Gaudet* does not describe, teach, or suggest a logically partition data processing system, storing trace data from all partitions in the same trace buffer, continually overwriting data that is already stored in the trace buffer with newly received data, storing error data associated with the error in the trace buffer, or saving the current contents of the trace buffer prior to error data in the buffer being overwritten. Therefore, the cited prior art does not render Applicants' claims unpatentable.

The Examiner rejected claims 12, 15, 27, 30, 42, and 45 under 35 U.S.C. § 103(a) as being unpatentable over *Pang* in view of *Gaudet* and further in view of Microsoft Dictionary, published 1999 [hereinafter *Microsoft*]. This rejection, as it might be applied to the claims as amended, is respectfully traversed.

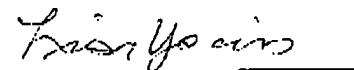
The Examiner states that the combination of *Pang* and *Gaudet* does not teach the use of an exception handler routine in the servicing of the error. The Examiner states that *Microsoft* teaches that exception handling is a situation in which a separate routine must be used to deal with a problem in the conditions of the system. The combination of *Pang* and *Gaudet* does not teach a logically partition data processing system, storing trace data from all partitions in the same trace buffer, continually overwriting data that is already stored in the trace buffer with newly received data, storing error data associated with the error in the trace buffer, or saving the current contents of the trace buffer prior to error data in the buffer being overwritten. Therefore, the combination of *Pang*, *Gaudet*, and *Microsoft* does not teach these features in combination with an exception handler routine.

The cited references do not describe, teach, or suggest a logically partition data processing system, storing trace data from all partitions in the same trace buffer, continually overwriting data that is already stored in the trace buffer with newly received data, storing error data associated with the error in the trace buffer, or saving the current contents of the trace buffer prior to error data in the buffer being overwritten. Therefore, the cited references do not render Applicants' claims unpatentable.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: 12.28.04

Respectfully submitted,



Lisa L.B. Yociss  
Reg. No. 36,975  
Yee & Associates, P.C.  
P.O. Box 802333  
Dallas, TX 75380  
(972) 385-8777  
Attorney for Applicants